PETSc: An Advanced Math and Computing Framework for Rapidly Developing Parallel Smart Grid Applications

Shrirang Abhyankar
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL
Email: abhyshr@mcs.anl.gov

Barry Smith
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL
Email: bsmith@mcs.anl.gov

***Abstract*—Developing parallel (scalable) software for smart grid applications is a challenging task entailing considerable time and effort. This paper introduces the open source high-performance library PETSc (Portable Extensible Toolkit for Scientific computation) as a potential math and computing framework for rapid development of parallel smart grid applications. PETSc has been tested on a gamut of scientific applications on single-core machines to supercomputers, has highly optimized implementations, and includes a wide array of tested numerical solvers. We provide an overview of PETSc's mathematical and computing capabilities and discuss a few emerging power system applications that can be developed by using PETSc.**

***Index Terms*—Power System Simulation, High Performance Computing, PETSc, Smart Grid**

I. Introduction

Smart grids bring the promise of greater reliability, control, and environmental awareness through various initiatives such as increased renewable generation, denser interconnecton between utilities, enhanced communication channels, and better insight into load diversity and characteristics. The penetration of wind, solar, and other renewable resources of electricity production is increasing. The advent of deregulation is driving the power industry toward economic operation and thus operating the transmission system to its fullest potential. In order to manage the load growth and to enhance reliability and security, the interconnection between utility controlled transmission systems is growing. The incorporation of power electronics controllers, an integral part of renewable energy and distributed energy resources (DER), will likely be increasing. As interconnections continue to grow, the need may arise for managing large-scale and ultra-large-scale transmission systems, whether regional, national, or multinational, in real time. These developments are making power system computational problems larger, denser, and multiscale in nature requiring high-fidelity models, faster simulation requirements, and need to deal with uncertainties.

A natural way to speed this computation is to use parallel computing techniques, namely, share the computational load among multiple processors. The need for parallel smart grid applications is even greater as the computer hardware industry moves multicore and many-core architectures. All major computer vendors are aggressively introducing a new generation of hardware that incorporates multiple cores on a chip, sometimes with simultaneous multithreading capabilities. Products incorporating 6 and 8 cores are already on the market. The number of cores per chip is expected to grow rapidly, so that even in the relatively short term, a single chip is expected to support the execution of a few hundred threads. These multicore architectures can be utilized efficiently only with parallel algorithms that distribute the computational load over multiple cores. Several workshops [1], [2] have highlighted the need for investigating these multicore/many-core architectures for accelerating performance of power system applications.

In this paper, we present the high performance computing library PETSc (Portable, Extensible Toolkit for Scientific computation) that can aid in rapid development of parallel smart grid applications. The range of parallel linear, nonlinear, time-stepping solvers, abstract linear algebra object interfaces for writing user application codes, portability to different operating systems, and flexible runtime options for faster experimentation, make PETSc an attractive math and computing framework.

II. PETSc Introduction

PETSc is a suite of data structures and routines that provide the building blocks for the implementation of large-scale application codes on parallel (and serial) computers. It is a part of U.S. Department of Energy's Advanced Computational Software [3] collection and was the winner of a 2009 R&D 100 award [4]. PETSc has been used for developing applications in various scientific fields, such as nanosimulations [5], [6], fusion [7], [8], power systems [9]–[11], and others [12].

PETSc is organized hierarchically, enabling users to employ the level of abstraction that is most appropriate for a particular problem. By using object-oriented programming, PETSc provides enormous flexibility for users. PETSc consists of a variety of libraries (similar to classes in C++); see Fig. 1. Each

Invited Paper for the TF on OSS: "Open Source Tools for Smart Grid Applications", IEEE PES General Meeting 2013.

library manipulates a particular family of objects (for instance, vectors) and the operations one would like to perform on the objects.

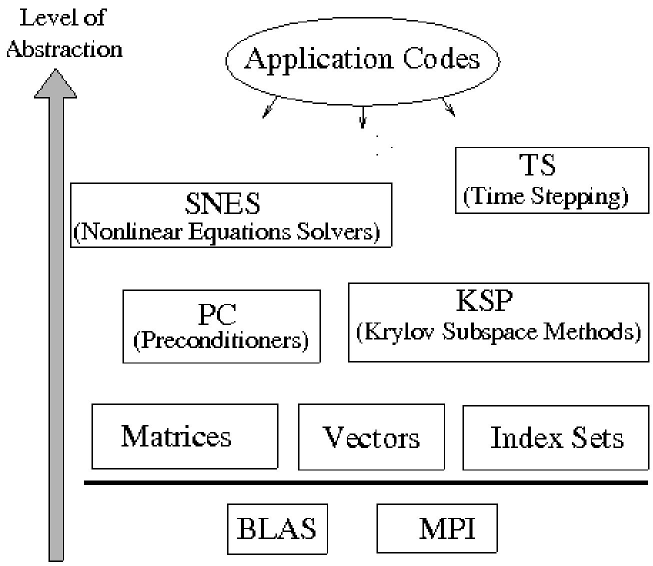

Fig. 1. Organization of the PETSc library [13]

Each object consists of an abstract interface (simply a set of calling sequences) and one or more implementations using particular data structures. Thus, PETSc provides clean and effective codes for the various phases of solving applications, with a uniform approach for each class of problems. This design enables easy comparison and use of different algorithms (for example, to experiment with different Krylov subspace methods, preconditioners, or truncated Newton methods). The libraries enable easy customization and extension of both algorithms and implementations. This approach promotes code reuse and flexibility and separates the issues of parallelism from the choice of algorithm. As a result, PETSc provides a rich environment for modeling large-scale scientific applications as well as for rapid algorithm design and prototyping.

III. Using PETSc and Supporting Libraries for Developing Smart Grid Applications

This section describes a few smart grid applications that can be developed by using PETSc.

A. Power Flow related applications

Power flow is a fundamental application in power system analysis. Various analyses such as steady state security, area power transfer studies, contingency screening, require a power flow solution. Essentially, the power flow problem solves the nonlinear power balance equation for the network given a generation set point and a load injection,

$$F(x) = 0 \tag{1}$$

where x are the bus voltages. In addition, a power flow solution serves as a starting point for dynamic simulations or short circuit studies. Faster solution of power flow equations would speed power system analysis programs entailing repeated power flow solutions such as contingency analysis, voltage stability studies.

TABLE I
List of Power System Applications that can be developed using PETSc and child libraries

Component/Library	Use	Applications
TS	Time-stepping algorithms	Transient Stability, Electromagnetic transients simulation, combined transient stability electromagnetic transients analysis
SNES	Nonlinear solver	AC power flow, voltage Stability analysis, contingency analysis
KSP and PC	Linear solver and preconditioning	DC power flow, shift factor calculation
TAO [14]	Convex Optimization	optimal power flow, security constrained economic dispatch
SLEPc [15]	Eigen value analysis	Small signal stability analysis

The nonlinear solver class SNES can be used for developing parallel power flow applications where the user needs to only provide a routine for the nonlinear function evaluation (and an optional Jacobian evaluation). Reference [11] discusses very large power flow analysis using an Inexact Newton-Krylov solver preconditioned with an incomplete LU factorization scheme.

B. Combined electromechanical-electromagnetic transients simulation

The number of power electronics devices is expected to increase for more flexible control of power systems. Therefore, the effect of non-fundamental frequency harmonics will increase, and the electromechanical dynamics simulation will entail modeling of power electronics devices by an electromagnetic simulation. An attractive way of incorporating the simulation of non-fundamental frequency harmonics in an electromechanical transient simulator is via a hybrid simulation [16]. In a hybrid simulation, most of the bulk power system is modeled by using an electromechanical transients simulator, while a small part of it is modeled by using an electromagnetic transients simulator. The PETSc library provides efficient data structures to ease the development of such multiscale or multiphysics applications. The details of an implicitly coupled electromechanical-electromagnetic simulation using the PETSc library are given in [17], and [10].

C. Combined transmission-distribution analysis

Various ISOs have indicated the need to model and gather real-time information from the subtransmission and distribution systems in order to provide finer-granularity load modeling [18]. While ISOs have traditionally been able to forecast load within a 2% error, deployment of distributed energy resources and utility-scale storage may increase the error substantially [18]. The sheer volume of the components for a combined transmission-distribution analysis presents an onerous computational task and emphasizes the need for developing parallel algorithms for such an analysis and the use of high-performance computing libraries.

D. Electromechanical transients simulation

Considerable research on parallel implementation of transient stability applications has been done by power system researchers [19]–[23], as it offers a possibility of real-time dynamic simulation. The TS library in PETSc can be used for efficiently solving nonlinear differential algebraic transient stability equations. TS provides various DAE solvers including the implicit-trapezoidal integration scheme. The recently released version of PETSc has added implicit-explicit (IMEX) time integration schemes for multirate problems that can also be experimented with.

Schur-complement is one of the preferred methods for solving the linearized system of transient stability equations. PETSc's multiphysics Schur-complement based solver [24] can be selected at run-time by specifying sets of indices for the generator and the network blocks. Various conjugate-gradient-based algorithms (conjugate gradient, conjugate gradient square, bi-conjugate gradient) are also available and can be selected at runtime.

Reference [25] presents parallel power grid dynamics simulation using the PETSc library. The scalability results, shown in Fig. 2, with a block-Jacobi preconditioned Newton-GMRES scheme are compared with parallel direct LU factorization using the MUMPS [26] package.

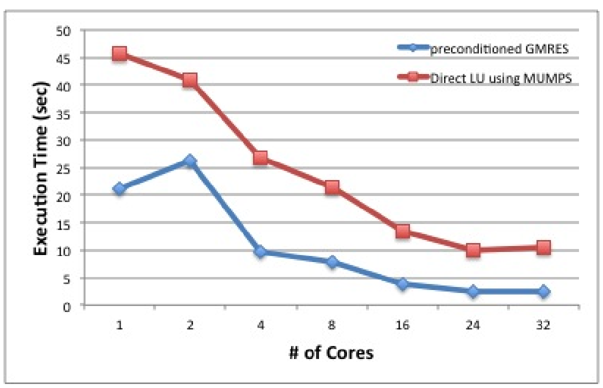

Fig. 2. Comparison of execution times using different linear solution schemes for parallel dynamics simulation of a test 2383-bus system dynamics subjected to a balanced three-phase fault for 0.1 seconds [25].

E. Electromagnetic Transients Simulation

The ultimate goal for the power system simulation researchers is an electromagnetic transient simulation in real-time at the microsecond level. However, the modeling complexity along with time-step limitations for electromagnetic simulations are overwhelming. The transmission lines for electromagnetic transients simulation are modeled by traveling wave equations consisting of two disjoint equivalent circuits. This model is nicely structured for parallel processing with equations for a geographical subsystem being assigned to each processor.

F. Power system optimization

PETSc has been developed primarily for solving linear and nonlinear algebraic equations and hence has limited support for optimization applications. The scalable optimization library TAO (Toolkit for Applied Optimization) [14], which is built using PETSc, can be used for convex power system optimization applications, for e.g., optimal power flow, and security constrained optimal power flow. Support for solving security constrained unit commitment applications needing mixed integer linear/nonlinear solvers has not yet been developed in PETSc or TAO.

G. Small signal stability

SLEPc (Scalable library for Eigenvalue Problem Computations) [15], a library based on PETSc, can be used for developing small signal stability analysis applications. SLEPc consists of several algorithms for scalable computation of eigenvalues and uses various PETSc data structures such as matrix storage schemes, and vectors.

IV. PETSc Math Capabilities

In this section, we describe the solver capabilities of PETSc by briefly introducting the key libraries.

A. Scalable Time-stepping integrators

The Time-stepping (TS) library in PETSc provides a framework for the scalable solution of systems of ordinary differential equations of the form $\dot{x} = f(x(t))$, differential algebraic equations of the form $F(x, \dot{x}, t) = 0$, and steady-state problems using pseudo-time stepping. The library contains various numerical integration algorithms such as explicit and implict Euler schemes; Crank-Nicholson; and multi-stage explicit adaptive Runge-Kutta and Rosenbrock-W variable time stepping schemes. Additionally, the sundials library [27], which includes variable-order, variable coefficient backward differential (BDF) differential-algebraic integrators, can be used with PETSc. The TS library uses PETSc's nonlinear solver library SNES and the linear solver KSP objects to solve the underlying nonlinear/linear system, and the user can tune this solution process at runtime, too.

B. Scalable Nonlinear solvers (SNES)

The nonlinear solver class SNES includes methods for solving systems of nonlinear equations of the form

$$F(x) = 0. \tag{2}$$

Newton-like methods provide the core of this library, including both line-search and trust-region techniques. PETSc also includes new capabilities for composing scalable nonlinear solvers such as nonlinear GMRES, nonlinear conjugate gradient, quasi-Newton, and nonlinear multigrid [28]. A new addition to the PETSc library is support for handling box-constrained algebraic variational inequalities, $x_{min} \leq f(x) \leq$

x_{max}. These algebraic variational inequalities arise as a result of a particular physical quantity hitting a hard limit, for e.g., over-excitation limiters for the generator automatic voltage regulator system. Built on top of the linear solvers and data structures, SNES enables easy customization of nonlinear solvers for the application at hand.

C. Linear solvers and preconditioners

PETSc provides scalable, composable linear solvers that can be tuned based on the application and can be changed at runtime.The linear solver library called KSP is the heart of PETSc: it provides uniform and efficient access to all of the package's linear system solvers, including parallel and sequential, direct and iterative. KSP is intended for solving nonsingular systems of the form

$$Ax = b, \tag{3}$$

where A denotes the matrix representation of a linear operations, b is the right hand side vector, and x is the solution vector.

The combination of a Krylov subspace method and a preconditioner is at the center of most modern numerical codes for the iterative solution of linear systems. Since the rate of convergence of Krylov projection methods for a particular linear system strongly depends on its spectrum, preconditioning is typically used to alter the spectrum and hence accelerate the convergence rate of iterative techniques. Currently, PETSc supports over 20 KSP methods and preconditioners. PETSc also allows composition of available preconditioners for creating new preconditioners. Solvers and preconditioners can be also nested. For example, with a parallel block-Jacobi preconditioner (i.e., the preconditioner formed using the diagonal block of the matrix on each processor), any of the other preconditioners, such as LU, ILU, or SOR, can be used on the block.

Various reordering schemes to reduce the fill-in for the factored matrices are also available and can be accessed either by calling routines or by using runtime options.

D. Vectors

The vector (denoted by Vec) is one of the simplest PETSc objects. Vectors are used to store solutions, right-hand sides for linear systems, and so forth. PETSc currently provides several basic vector types; the two most commonly used are sequential and parallel (MPI-based). Basic vector operations, such as dot product and sum, are available in the PETSc vector library. The comprehensive list of vector operations can be found in [29].

E. Matrices

PETSc provides a variety of matrix implementations because no single matrix format is appropriate for all problems. Currently PETSc supports dense storage and compressed sparse row storage, as well as several specialized formats such as block compressed sparse row and symmetric compressed and block compressed formats. Most matrix formats in PETSc support sequential (matrix values stored on single processor) and parallel (matrix values divided onto multiple processors) versions. An interface for adding user-defined matrix formats is also provided.

V. PETSc Computing Capabilities

Along with the range of numerical solvers, PETSc provides a portable and extensible platform for developing applications. PETSc's computing capabilities are briefly highlighted in this section.

- PETSc is an open source package for the numerical solution of large-scale applications and is free for anyone to use (BSD-style license), including industrial users.
- Application codes can be written in C, C++, Fortran, Python, or Matlab.
- The library provides multithreading (Pthreads, OpenMP) support for applications on shared-memory nodes.
- PETSc provides support for solving applications on NVidia GPUs.
- It is portable to various operating systems such as Linux, Microsoft Windows, Apple Macintosh, and Unix and can be also used from within the Microsoft Developers Studio. This portability aspect is important if the operating systems in different control centers are different.
- PETSc can be configured to work with real or complex data types (not mixed though), single or double precision, and 32 or 64-bit integers. It has been tested on a variety of tightly coupled parallel architectures such as Cray XT/5, Blue Gene/P, and Earth Simulator and also on loosely coupled architectures such as networks of workstations.
- Over 20 third-party open source high-performance computing packages can be used as a plug-in with PETSc. PETSc provides an interface for these external packages so that they can be used in application codes. PETSc can also download and install these packages along with the PETSc installation. A complete list of the external packages that can be used with PETSc can be found in [30]. This extensibility feature of PETSc allows exploration of other scalable libraries if need be.
- PETSc allows users to modify parameters and options easily at runtime. For example, users can modify the linear solution scheme from GMRES to direct LU factorization or can change the matrix storage type, or preconditioners, by using runtime options. If an application uses a large number of parameters, these can be supplied by a text file that is read when the PETSc code begins. Such flexibility is essential for applications that need to explore and compose the solution algorithms rapidly.

VI. Conclusions

Developing scalable applications is necessary as power systems expand, interconnection gets denser, and newer equipment gets added. This paper discussed the high-performance computing library PETSc as a potential platform for rapid development of parallel smart grid applications. The development of PETSc has been funded by the U.S. Department of Energy for over 15 years. Because of its wide use among U.S. DOE application scientists, its continued long term development and support are highly likely.

ACKNOWLEDGMENTS

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Dept. of Energy, under Contract DE-AC02-06CH11357.

REFERENCES

[1] J. H. Eto and R. J. Thomas, "Computational needs for the next generation electric grid," U.S. Department of Energy, Tech. Rep., 2011, http://energy.gov/sites/prod/files/FINAL_CompNeeds_Proceedings2011.pdf.

[2] E. P. R. Institute, "Grid transformation workshop results," Electric Power Research Institute, Tech. Rep., 2012, http://my.epri.com/portal/server.pt?space=CommunityPage&cached=true&parentname=ObjMgr&parentid=2&control=SetCommunity&CommunityID=404&RaiseDocID=000000000001025087&RaiseDocType=Abstract_id.

[3] "Advanced Computational Software (ACTS) Web page," http://acts.nersc.gov.

[4] R&D Magazine, "PETSc R&D 100 award," 2009, see http://www.rdmag.com/Awards/RD-100-Awards/2009/07/PETSc-Release-3-0-Expands-Capabilities.

[5] L. Yuan, J. Jiang, G. Du, Z. Wang, and H. He, "Micromagnetic simulation of magnetic particles with surface anisotropy," *Journal of the Magnetics Society of Japan*, vol. 32, pp. 50–53, 2008. [Online]. Available: http://www.jstage.jst.go.jp/article/msjmag/32/2_1/32_50/_article

[6] S. Greaves, "Micromagnetic simulations of magnetic recording media," in *High Performance Computing on Vector Systems 2007*, 2007. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-74384-2_17

[7] J. Cary, J. Candy, J. Cobb, R. H. Cohen, T. Epperly, D. Estep, S. Krasheninnikov, A. Malony, D. McCune, L. McInnes, A. Pankin, S. Balay, J. Carlsson, M. R. Fahey, R. Groebner, A. Hakim, S. Kruger, M. Miah, A. Pletzer, S. Shasharina, S. Vadlamani, D. Wade-Stein, T. Rognlien, A. Morris, S. Shende, G. W. Hammett, K. Indireshkumar, A. Pigarov, and H. Zhang, "Concurrent, parallel, multiphysics coupling in the FACETS project," *J. Phys.: Conf. Ser.*, vol. 180, p. 012056, 2009. [Online]. Available: http://iopscience.iop.org/1742-6596/180/1/012056

[8] B. Philip, M. Pernice, and L.Chacon, "Solution of reduced resistive magnetohydrodynamics using implicit adaptive mesh refinement," in *Proceedings of the 16th International Conference on Domain Decomposition methods*, 2005.

[9] S. Abhyankar, B. Smith, H. Zhang, and A. Flueck, "Using PETSc to develop scalable applications for next-generation power grid," in *Proceedings of the 1st International Workshop on High Performance Computing, Networking and Analytics for the Power Grid*. ACM, 2011. [Online]. Available: http://www.mcs.anl.gov/uploads/cels/papers/P1957-0911.pdf

[10] S. Abhyankar and A. Flueck, "An implicitly coupled solution approach for combined electromechanical and electromagnetic transients simulation," in *Proceedings of the IEEE Power and Energy Society General Meeting*. IEEE, 2012.

[11] R. Idema, D. Lahaye, C. Vuik, and L. van der Sluis, "Scalable newton-krylov solver for very large power flow problems," *IEEE Transactions on Power Systems*, vol. 27, pp. 390–396, 2012.

[12] B. Smith et al., "Scientific Applications Using PETSc," http://www.mcs.anl.gov/petsc/publications.

[13] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. D. Gropp, D. K. añd Matthew G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, "PETSc users manual," Argonne National Laboratory, Tech. Rep. ANL-95/11 - Revision 3.2, 2011. [Online]. Available: http://www.mcs.anl.gov/petsc

[14] T. Munson, J. Sarich, S. Wild, S. Benson, L. C. McInnes, and J. Moré, "Toolkit for Advanced Optimization (TAO) Web page." [Online]. Available: http://www.mcs.anl.gov/tao

[15] C. Campos, J. E. Román, E. Romero, A. Tomás, V. Hernández, and V. Vidal, "SLEPc homepage," scalable Library for Eigenvalue Problem Computations. [Online]. Available: http://www.grycap.upv.es/slepc/

[16] IEEE Task Force on Interfacing Techniques for Simulation Tools, "Interfacing techniques for transient stability and electromagnetic transients program," *IEEE Transactions on Power Apparatus Systems*, vol. 8, pp. 2385–2395, 2009.

[17] S. Abhyankar, "Development of an implicitly coupled electromechanical and electromagnetic transients simulator for power systems," Ph.D. dissertation, Illinois Institute of Technology, 2011.

[18] S. Grijalva, "Research needs in multi-dimensional, multi-scale modeling and algorithms for next generation electricity grids," in *DOE workshop on computing challenges for the next-generation power grid*, 2011.

[19] I. Decker, D. Falcao, and E. Kaszkurewicz, "Parallel implementation of power system dynamic simulation methodology using the conjugate gradient method," *IEEE Transactions on Power Systems*, vol. 7, pp. 458–465, 1992.

[20] J. Chai and A. Bose, "Bottlenecks in parallel algorithms for power system stability analysis," *IEEE Transactions on Power Systems*, vol. 8, pp. 9–15, 1993.

[21] F. Alvarado, "Parallel solution of transient problems by trapezoidal integration," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-98, pp. 1080–1090, 1979.

[22] M. Ilic, M. Crow, and M. Pai, "Transient stability simulation by waveform relaxation methods," *IEEE Transactions on Power Systems*, vol. 2, pp. 943–952, 1987.

[23] M. Crow and M. Ilic, "The parallel implementation of waveform relaxation methods for transient stability simulations," *IEEE Transactions on Power Systems*, vol. 5, pp. 922–932, 1990.

[24] B. Smith, L. C. McInnes, E. Constantinescu, M. Adams, S. Balay, J. Brown, M. Knepley, and H. Zhang, "PETSc's software strategy for the design space of composable extreme-scale solvers," Argonne National Laboratory, Preprint ANL/MCS-P2059-0312, 2012, DOE Exascale Research Conference, April 16-18, 2012, Portland, OR.

[25] S. Abhyankar and A. Flueck, "Real-time power grid dynamics simulation using a block-jacobi preconditioned newton-gmres scheme," in *To appear in the Proceedings of the 2nd International Workshop on High Performance Computing, Networking and Analytics for the Power Grid*. IEEE, 2012.

[26] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster "A fully asynchronous multifrontal solver using distributed dynami c scheduling," *SIAM Journal on Matrix Analysis and Applications*, vol. 23, no. 1, pp. 15–41, 2001.

[27] C. Woodward et al., "SUNDIALS Web page," https://computation.llnl.gov/casc/sundials/main.html.

[28] P. Brune, M. Knepley, B. Smith, and X. Tu, "Composing scalable nonlinear solvers," Argonne National Laboratory, Preprint ANL/MCS-P2010-0112, 2012.

[29] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, "PETSc Web page," http://www.mcs.anl.gov/petsc, 2011. [Online]. Available: http://www.mcs.anl.gov/petsc

[30] B. Smith et al., "External Software Used by PETSc," http://www.mcs.anl.gov/petsc/miscellaneous/external.html.